

$esABB(t) \Rightarrow esABB(insABB \ x \ t)$

$p.t = esABB(t) \Rightarrow esABB(insABB \ x \ t)$

### **Caso Base**

$p.Null$

$p.Null$   
= $\{def \ p\}$   
 $esABB(Null) \Rightarrow esABB(insABB \ x \ Null)$   
= $\{def \ insABB\}$   
 $esABB(Null) \Rightarrow esABB(Node \ Null \ x \ Null)$   
= $\{def \ esABB\}$   
 $True \Rightarrow True$   
= $\{reflex \ implicación\}$   
 $True$

### **Hipótesis Inductiva**

$p.hi$   
= $\{def \ p\}$   
 $esABB(hi) \Rightarrow esABB(insABB \ x \ hi)$   
= $\{def \ esABB\}$   
 $listOrd(treeToList \ hi) \Rightarrow listOrd(treeToList \ (insABB \ x \ hi))$

$p.hd$   
= $\{def \ p\}$   
 $esABB(hd) \Rightarrow esABB(insABB \ x \ hd)$   
= $\{def \ esABB\}$   
 $listOrd(treeToList \ hd) \Rightarrow listOrd(treeToList \ (insABB \ x \ hd))$

### **Etapla inductiva**

$esABB(Node \ hi \ r \ hd) \Rightarrow esABB(insABB \ x \ (Node \ hi \ r \ hd))$   
= $\{def \ insABB\}$

**$x=r$**

$esABB(Node \ hi \ r \ hd) \Rightarrow esABB(insABB \ x \ (Node \ hi \ r \ hd))$   
= $\{def \ insABB\}$   
 $esABB(Node \ hi \ r \ hd) \Rightarrow esABB(Node \ hi \ r \ hd)$   
= $\{reflex \ implicación\}$   
 $True$

**$x<r$**

```

    esABB(Node hi r hd)  $\Rightarrow$  esABB(insABB x (node hi r hd))
={def insABB}
    esABB(Node hi r hd)  $\Rightarrow$  esABB(Node (insABB x hi) r hd)
={def esABB}
    listOrd(treeToList(Node hi r hd))  $\Rightarrow$  listOrd(treeToList(Node (insABB x hi) r hd))
={def treeToList}
    listOrd(treeToList hi ++ [r] ++ treeToList hd)  $\Rightarrow$  listOrd(treeToList (insABB x hi) ++ [r] ++
treeToList hd)
={ [r] ++ treeToList hd esta ordenado por el antecedente
    treeToList (insABB x hi) esta ordenado por HI p.hi
    como x<r y (insABB x hi) esta ordenado, entonces todo esta ordenado y si esta
ordenado esABB}
True

```

**x>r**

```

    esABB(Node hi r hd)  $\Rightarrow$  esABB(insABB x (node hi r hd))
={def insABB}
    esABB(Node hi r hd)  $\Rightarrow$  esABB(Node hi r (insABB x hd))
={def esABB}
    listOrd(treeToList(Node hi r hd))  $\Rightarrow$  listOrd(treeToList(Node hi r (insABB x hd)))
={def treeToList}
    listOrd(treeToList hi ++ [r] ++ treeToList hd)  $\Rightarrow$  listOrd(treeToList hi ++ [r] ++ treeToList
(insABB x hd))
={treeToList hi ++ [r] esta ordenado por el antecedente
    treeToList (insABB x hd) esta ordenado por HI p.hd
    como x>r y (insABB x hd) esta ordenado, entonces todo esta ordenado y si esta
ordenado esABB}
True

```